

DOCUMENT RESUME

ED 244 555

HE 017 258

AUTHOR Klingenstein, Kenneth J.
TITLE Academic Support Systems: A Networked Approach.
PUB DATE Dec 83
NOTE 12p.; In: Information Resources and the Individual. Proceedings of the CAUSE National Conference (San Francisco, CA, December 11-14, 1983). Boulder, CO, CAUSE, 1984, p. 165-175. For the complete proceedings, see HE 017 245.
PUB TYPE Viewpoints (120) -- Speeches/Conference Papers (150)
EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS *Academic Deans; *Administration; *College Administration; College Faculty; *Computer Oriented Programs; Economic Factors; Higher Education; *Information Systems; Political Influences; Program Implementation; Services; *Systems Analysis; Technology
IDENTIFIERS University of Colorado Colorado Springs

ABSTRACT

Basic design and implementation issues encountered in providing academic support systems are examined. "Academic support systems" describes the growing class of university computing that is being developed to integrate administrative computing environments. These systems may be typically characterized by the activities that center around a dean's office: office automation/information systems, institutional research, and electronic communication. The components (and their interfaces) of academic support systems are discussed, distinguishing the activities that are more properly considered in strictly academic or administrative arenas. After examining the basic design issues that a system planner faces, the generic academic support architectures are analyzed with respect to political, financial, and technical factors. These issues include speed of links and compatibility requirements (technical), licensing of software and billing of network services (financial), and user education and data ownership (political). To illustrate the issues, the implementation of an academic support system at the University of Colorado, Colorado Springs is described. (SW)

* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *

ACADEMIC SUPPORT SYSTEMS: A NETWORKED APPROACH

Dr. Kenneth J Klingenstein
University of Colorado at Colorado Springs
Colorado Springs,
Colorado 80907

Academic support systems describes the growing class of university computing that are being developed to integrate and exchange information from both the academic and administrative computing environments. These systems may be typically characterized by the activities that center around a dean's office - office automation/information systems, institutional research, and electronic communication. This paper examines the basic design and implementation issues that are encountered in providing these support systems. The generic architectures are evaluated for their technical, financial and political strengths and weaknesses. Common implementation concerns such as security and telecommunications backbone are discussed.

"PERMISSION TO REPRODUCE THIS
MATERIAL HAS BEEN GRANTED BY

CAUSE

TO THE EDUCATIONAL RESOURCES
INFORMATION CENTER (ERIC)."

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION
CENTER (ERIC)

- ☐ This document has been reproduced as received from the person or organization originating it.
- ☒ Minor changes have been made to improve reproduction quality.
- Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

Paper presented at the
CAUSE National Conference
(San Francisco, CA, Dec. 11-14, 1983)

ED244555

NE 017 258

One of the most important trends in university computing is the development of systems to manage information drawn from both the administrative and academic computing sectors. These connective systems, called academic support computing, can be characterized by a spectrum of services such as faculty scoring, computing, and registering grades from their home microcomputers, deans seeking statistical analysis of student population, and budget offices coordinating grant budgets with researchers. Reflective of a more managerial style in academic administration, this intersection of administrative and academic systems is a critical and difficult area, if only because of the clientele being served and the multiple interfaces required.

There have been several excellent reports on individual institution's approaches to providing academic support services, but there are few papers on ordering possible approaches within the broader framework of the institution's academic and administrative systems. This paper will focus on the generic design issues that need to be addressed by the organization providing the academic support services and how the rest of the institution's computing environments help shape the implementation. It is hoped that such a structured analysis may aid those who are also wrestling with the unique demands of this type of computing. While one specific architecture, a hierarchy of micros, minis and mains, is discussed below, it should be stressed that this solution emerged as best due only to a particular local computing environment.

The paper is divided into four sections. Section I discusses the components (and their interfaces) of academic support systems, distinguishing the activities that are more properly considered in strictly academic or administrative arenas. We also look at the underlying political, economic, and developmental needs that support systems must address. In section II, we examine the basic design issues that a systems planner faces. The generic academic support architectures are then analyzed with respect to these political, financial and technical factors. Implementation concerns are surveyed in section III. These issues are technical (e.g. speed of links, compatibility requirements), financial (e.g. licensing of software, billing of network services) and political (e.g. user education, data ownership) stuff. Section IV presents some concluding remarks.

WHAT ARE ACADEMIC SUPPORT SYSTEMS?

The growth of university academic and administrative computing systems in both power and scope has produced a growing intersection of these activities. The intersection is marked not by shared activities

but by shared data. Computations remain distinct; instructional and academic research computing continues to be the province of academic computing, and payroll and finance remain in the administrative domain. (In this paper, all these computations are grouped as the "primary systems", to reflect both their importance and their history.) However users of one primary system are increasingly accessing or wanting features on the other primary system. The prototypic situation is a dean's office (or associate deans on larger campuses). Almost every tool required in this office is becoming automated. Paperwork is handled by word processors and database or spreadsheet applications. Institutional research is conducted through downloading of administrative databases, often to academic systems which have the packages to do the statistics. Perhaps the most critical activity for the Dean is coordinating between academics and administrators; the considerable communication required is increasingly being performed electronically.

Other examples of the need for academic support systems abound. A faculty member familiar with electronic mail between his colleagues wants the same ability to communicate to administrative units when he assumes an administrative position. Crossover also occurs when a staff member takes a class and wants to use the administrative terminal on their desk to access the academic system for their assignments. These examples illustrate a key aspect of this type of demand - knowledgeable users making sophisticated but reasonable requests.

Three broad classes of academic support services have emerged. They may be loosely labeled as: office automation/information systems, institutional research, and communications. The first category includes many of the "convenience computing" activities that are becoming prevalent, such as word processing, electronic spreadsheets, and the myriad office applications that respond so well to simple database packages. The most important aspect of this class for the academic support designer is the local origination and ownership of the datasets generally utilized in these situations. Such data poses less security and compatibility constraints to the rest of the systems. Another significant feature is that many of the applications are user-developed; appropriate central support is discussed below.

The increasing importance of the second category, institutional research, is indicative of the more managerial tone current in university administration. Calculations such as the relative cost of educating a student in different disciplines, comparative salaries by department, or the projected mix of students in 1988 are valuable statistics in planning and managing the modern university. Institutional research can be characterized by the eclectic sources of its input data, its passive but frequent use of institutional databases (such as payroll or SIS - student information systems) and

the lack of need for its output data by the rest of the institution. For example, the data sources for determining educational cost per discipline includes budgetary databases, SIS, and capital inventories, while projections of student mix draws from such diverse sources as demographic studies, business employment trends, SIS, marketing analyses, etc. Where such research accesses university databases, it does so in read-only mode and often for only aggregated data. In addition, the results of the calculations are seldom accessible to other institutional units, so that "production" concerns do not apply. While security issues are somewhat more important in this category, local ownership and read-only accesses of institutional databases limits the vulnerability.

The third class of academic support systems is communications. There are many implementation of computer-assisted communication: electronic mail, digitized voice, voice store-and-forward, etc. Communications can be categorized by locus (intraoffice, interoffice, intercampus) with relative traffic ratios depending on factors such as campus geography, whether the school is part of a multicampus institution, and degree of national prominence. A second important view of communication is the time frame in which responses are needed, and a third major variable is the size of the average message. Such analyses are important in planning the communications and networking aspects of academic support systems.

Several recurring themes can be found in the above discussion. First, academic support systems is the intersection of two orthogonal computing styles. It is a clash of interactively-oriented academic computing and batch-oriented administrative systems, a traditionally more distributed environment versus a traditionally more centralized environment, an area the emphasis is on doing 'it' your own way meeting an area where the emphasis is on doing 'it' every day. Secondly, academic support systems seldom have a natural server. It does not belong purely on either the academic or administrative hardware, nor is it clear just who should manage this rather sensitive service. Thirdly, growing as an interface to existing academic and administrative systems, academic support systems must usually be retrofitted onto a diverse and often arcane set of compatibility conditions. Retrofitting is always a difficult task, and it is hard to design a powerful and flexible system that must conform to some archaic but required architecture.

DESIGN OF ACADEMIC SUPPORT SYSTEMS

The design of academic support systems takes place in a highly constrained and complicated environment. Typical difficulties include retrofitting to existing systems, the need for significant capitalization to provide the hardware for the services, and the politics of a visible and sophisticated clientele. We will examine some of these design issues in their technical, financial and political aspects; the mix of factors present in any situation of course is quite dependent on the institution.

Technical

Most of the technical design issues relate to the dual problems of providing the academic support systems vehicles and retrofitting these to the existing academic and administrative computing systems. Unless the academic and administrative systems share a common machine, the choice of a hardware configuration and the development of avenues back into the primary systems are the central decisions; secondary concerns include the telecommunications backbone, the degree and type of compatibility between user workstations, data ownership, and security mechanisms.

There are three basic architectures for academic support systems: embedding the services on either (or both) academic and administrative systems; using a separate 'mainframe' with interfaces to the primary systems and terminals to the user desks; and, at the far end of the spectrum, a completely distributed system where user stations are autonomous nodes on some network that includes the primary systems. Sharing hardware with existing systems is the easiest option technically, but few computing systems will admit to the idle capacities that the support systems will require. Using a mainframe (by mainframe we mean a large word machine capable of supporting a reasonable number of time-shared users) and terminals presents a straightforward technical approach. Total interfaces to existing systems is kept to a minimum, providing both security and reduced development difficulties. However, academic support system data is not kept locally, increasing other security issues. Similarly, compatibility of devices (e.g. printers, crts, etc.) and software is almost assured, though in general at some cost in flexibility to individual unit's needs. Communications capabilities are also facilitated by this architecture, and institutional data held by the support systems can be limited to a single copy, reducing redundancy (dangerous to the integrity of databases) and storage costs. At the other end of the spectrum is a completely distributed network of user workstations, typically microcomputers. Data is kept locally,

securely, and often redundantly. Communications can be extremely difficult, especially with a variety of systems, posing a n^2 solution. (It is unusual that even off-the-shelf communication systems such as ethernet will have identical implementations on a diversity of systems.) Similarly, the number of interfaces to existing academic and administrative systems goes from 2 to $2n$, and security, compatibility difficulties, and complexity rises accordingly.

In practice, the weaknesses of the pure architectures can be resolved, and the strengths maintained, by use of a mixed architecture, where micros are netted through minis to the primary mainframes. As discussed below, such an implementation is flexible but with a high degree of compatibility and a limited number of interfaces to develop. The minis also provide the hub for communications.

The communications backbone must reach to every unit involved in the academic support system. The backbone may already exist if multiple site- multiple host multiplexing is used by either primary system, though security and traffic problems can result from piggybacking onto such a net. If there are local nets within buildings, then the backbone can consist of bridges from these nets into a central switch device. If a complete new backbone is needed, the designer will discover that, due to intersecting cost curves, telecommunication expenditures may easily exceed the cost of the systems being connected.

Financial

Regardless of the technical design, there are several common financial issues. The initial concern is the funding for the hardware, systems development and maintenance. Costs should be broken down into a one-time capitalization/system development and an ongoing system maintenance fee (which should include periodic system upgrades). Central systems offer uniform or pro rata cost schemes, while the distributed architectures offer the option of users purchasing their own workstations, which allows users to provide for their own needs. One particularly sensitive issue is whether new users be charged for initial system costs, and how new utilities are to be funded. Group licensing of software faces the same problem; the marginal cost to a new user is often less than the charges incurred by the first purchasers.

Financing the telecommunications backbone and the interfaces to the primary systems is a second set of issues. The designer must specify a uniform "demark" point where user costs and responsibilities begin. Charging at this entry point can be based on several factors, including prorata costs, speed of link, traffic volume by bits or messages. (Some ports, such as bulletin boards, have asymmetric

input/output volumes; whether one charges for input on output traffic is delicate.) To assure compatibility in communications, designers should specify a rigorous protocol that users must adhere to at the demark point. The level of the protocol depends on the architecture, but in general the specification should be to the transport layer of the ISO model.

Life-cycle cost analysis is deceptive in a field where prices are decreasing; it superficially seems worthwhile to continue to wait as prices to drop and performance improve. While replacements may pass a straight-forward cost-benefit analysis, the analysis for new functionality must attempt to calculate the lost opportunity costs of waiting for increased price/performance. Such calculations are always difficult to defend.

Political

Of course the academic support system must be designed within the political environment of the institution. The consequence of the clientele and the scope of the service increase the importance of the political design issues. Who provides the support service depends on the organization of the institution. Where one agency provides both the academic and administrative systems, the choice is obvious. In a multicampus institution with a centralized primary system, it is still desirable to provide the support systems locally. In a mature university with several fully developed departmental academic support systems, the role of a central provider (and the discrepancies between the haves and have-nots) must be carefully defined.

Because the users are generally sophisticated and knowledgeable, the opportunity should be taken to allow them to assume greater responsibility for their data and resources (such as backups and security) as well as application development where appropriate. Besides the obvious benefits of removing impediments to user power, such users tend to be more understanding and supportive of the central systems. Similarly, these users have the confidence and the exploratory nature that create successful user communities. User education programs are vital parts of academic support systems, and are often self-sustaining at this level.

IMPLEMENTATION ISSUES

Many implementation decisions follow directly from a well-done design. However there are some sets of issues best kept generic until system construction begins (e.g. speed of individual ports, location of shared letter-quality printers, etc.). In addition, RFPs may not come back as expected. Even worse is the computer analogue to the "round plug in a square socket" phenomena - when vendor performance specifications assume, but do not explicitly state, incompatible interface environments.

Viewing compatibility on a functional level leads to new interpretations of familiar terms. Hardware compatibility indicates that peripherals may be exchanged, offering a redundancy that is often valuable. Software compatibility means that a secretary trained on one work station can switch to another if needed. Data compatibility is perhaps the most critical in the academic support system environment with its propensity for information integration and exchange. It is increasingly important to be able to directly incorporate graphics or spreadsheets directly into text systems, to be able to use a familiar word processing editor while in electronic mail or to be able to consolidate spreadsheets across campus. Note that complete standardization on one text editor, one spreadsheet, one database manager, etc. gives no assurance of data compatibility or portability. Rather than face a combinatoric explosion of interfaces, it is best to standardize on a vanilla format (e.g. ASCII) and have each package provide an i/o utility to vanilla. (Of course, even vanilla has different flavors.) Data portability also requires an exchange medium to deal with incompatible disk formats, etc.; the communications support subsystem should have this capability.

Security is, of course, critical. It is also elusive in a university environment, with its high density of hackers. Ports into the primary administrative systems offer high vulnerability for the institution. One valuable technique is to restrict entry from the academic systems into the administrative systems to batch mode. Batch offers the best opportunities for security checks and journalling. Batch updates such as faculty entering final grades can be made to pass through critical filters in the academic support systems before entering the administrative system. Interactive read-only access to the administrative systems may be permitted, but more secure alternatives do exist; small high-priority batch utilities may be provided to fetch the data with little response degradation but increased scrutiny. In general the interfaces from the support systems into the academic systems may be fully interactive.

To illustrate these issues, we will describe the implementation of an academic support system at a specific institution, pointing out in parenthetical remarks the options typically encountered in such implementations.

A brief description of the institution and its computing environment: the University of Colorado at Colorado Springs (UCCS). Smallest (5000 headcount) of the four campus University of Colorado system. Traditional administrative systems (payroll, admissions, billing and receivables, etc.) provided centrally at the main campus at Boulder 120 miles away on an IBM mainframe. Academic services provided locally on networked DEC machines with a remote site terminal switching network to almost all buildings on a densely sited campus. No appreciable embedded base of departmental computers or word processors. Campus deans and directors are central figures with institutional research and governance both within the campus and between campuses. A single local computing services department provides the academic computing resources and operates the RJE and interactive terminus of the central administrative system.

This environment removed some design pitfalls (single computing organization, no embedded equipment bases to create inequities or incompatibilities) and exacerbated others (such as the DEC/IBM differences). The traditional weakness of the campus in word processing became a strength; hardware, software and data compatibility could be designed in, and word processing could be addressed by a multipurpose microcomputer. (To experienced word processors, the packages available on micros may be painfully primitive; to novices, however, it's sliced bread.) The central administrative systems provided economy and multicampus coherency.

A choice was made to build an academic support system through a micro, mini, and mainframe hierarchy accessible from a single keyboard. This architecture had several compelling virtues in the UCCS environment: Multipurpose office automation functionality for users who had no other equipment to handle any of these needs; major aspects of the funding assumed by users; flexibility; and decreased need for communication in a particularly costly telecommunications situation. The vendor choice on both the micros and the mini were logically limited by compatibility and staffing requirements to DEC and IBM; a DEC mini (VAX 750) was chosen because its compatibility to local systems, while micros have been confined to both DEC and IBM. (Departments may acquire other micros, but the demark point of other brands is the multiplexor port, while the IBM and DEC communications paths are maintained all the way to the user by the support coordinators.) Micros are being acquired by every top-line office down through all Deans and Directors.

The view into the micro's monitor reveals three distinct machines forming the academic support systems. Office automation and local information management are handled by the micro. Communications are run on the mini through terminal emulation mode on the micro; electronic mail extends to almost all academic and administrative areas both locally and on other campuses. Institutional research is done in several modes, one of which is the micro emulating an DEC VT100 which in turn emulates an IBM 3278 through a long-haul communications link from the mini to the central administrative system. The mini is also the RJE terminus of that long-haul link, so that batch jobs submitted to the central machine can return data to either the mini or the micro for research purposes.

Tuning the speed of the links between the micros and the mini led to several implementation options, including Ethernet and RS232. While the price of Ethernet has dropped rapidly, surprisingly the price of RS232 has dropped even more. Furthermore the traffic volume, while generally obeying the 80/20 rule, is a lesser consideration on a autonomous multipurpose microcomputer, where communication is just one of several activities. Lastly, the bursty but infrequent line activity meant that multiplexing RS232 could be done at a input-to-output aggregate speed ratio significantly higher than the standard 2:1. This higher ratio (at least 4:1) further improves the price/performance of RS232. The decision to use RS232 as the communications medium was termed "poor man's networking" but in retrospect turned out to be quite cost-effective in a system where file service is a secondary activity.

Equally important in implementation is the care and feeding of the academic support systems users. In general these users are extremely busy and have little "hacker" instincts; their interest is only in a tool that will save some labor. Like other novices, there is often a good deal of fear in these users; in addition they are usually confident and successful people who may have difficulty in admitting their fear (with the excuse "I'm just too busy to learn") and accepting instruction from a student assistant. These concerns should be acknowledged and user education should be focused in peer groups. Moreover there is an fallacious assumption by novice users that the machine, since it is billed as a labor-saving device, will give them significant amounts of free time. During the learning period (which may last as long as six months for an infrequent user) substantial effort will be required. Then, once past this period, the average user will find that as he can answer more questions with his system, he will naturally increase the applications he will tackle. The basic fact is that the user will do more because he can do more. Efficiency certainly improves, but that seldom translates into leisure time.

REMARKS

Several interesting observations have emerged from the design and implementation effort at UOCS. In our experience, we began by suggesting that users not buy hard disk systems, since hard disk price/performance is clearly the most rapidly changing aspect of micros. But the need for friendliness has revised this thinking; the complete menu approach without booting that is possible with hard disks is quite attractive to novices. Another lesson is that too many functions were embedded into a single keyboard. The lack of alternative equipment means constant competition for the multipurpose keyboard; either enough keyboards must be provided for all purposes or dedicated systems must also be acquired to offload some functionality from the academic support system keyboard. A third lesson is that portables and rentals are important to a successful implementation. Portables are extremely beneficial to a clientele that often have to bring work home. Rentals provided by the central coordinating agency may be necessary to bring critical but indigent offices into the communication net.

A second set of problems relate to the difficulties in maintaining interfaces to systems in which the academic support user is not the main user constituency. The primary systems constantly evolve in dynamics with their own users. The academic support systems must flex with these changes in an essentially passive relationship; it is politically unfeasible for the support systems to make demands on or resist changes of systems whose intent is to serve others.

Lastly, there are the caveats that all computing designers face these days: the twin evils of looking back (few have the perspicacity to fully perceive the current technological changes and their effects on users) or writing in paralysis for the patterns to emerge (the only pattern is change); and Murphy's Law, which is quite prevalent in a discipline where micron precision is critical but the discrepancy in so-called standards is rampant.

The role of academic support systems will certainly increase in the next few years. Meeting its difficult challenges will be a continuing theme of university computing.